# The case study: "No maximum at the R2-curve"

## Author

Olga Krasotkina for MPI

## Problem statement

In some cases the PR2 curve is monotonic and has not any extremum and sugessts very large optimal smoothness tending to the full stationary betas.

## Source data

There are 12 indicies an 158 funds for 60 time points from 2015-03-29 till 2015-06-12.

## Research steps

\* Done Python workspace for non-stationaty regression and cross-validation analysis. At this stage I use the non-stationaty regression estimation procedure without any constraints and \*\*leave-one-out\*\* algorithm as cross-validation. I would like to make research step-by-step from basic to complicated model

\*

## Conclusion

There are two problem in this data.

\* The return of I2 index at the 2015-03-25 is oulier.

\* \*\*"Zero-return"\*\* problem. The presence of $0$ in the data tend to the ambiguity of the betas choice. In this data where are a lot of zeros for the cash-like asset.

[](http://localhost:8888/tree)

R2\_research1 Last Checkpoint: 4 hours ago Autosave Failed!



Python 2

error

* File
* Edit
* View
* Insert
* Cell
* Kernel
* Help

CodeMarkdownRaw NBConvertHeading

CellToolbar

# The case study: "No maximum at the R2-curve"

## Author

Olga Krasotkina for MPI

## Problem statement

In some cases the PR2 curve is monotonic and has not any extremum and sugessts very large optimal smoothness tending to the full stationary betas.

## Source data

There are 12 indicies an 158 funds for 60 time points from 2015-03-29 till 2015-06-12.

## Research steps

\* Done Python workspace for non-stationaty regression and cross-validation analysis. At this stage I use the non-stationaty regression estimation procedure without any constraints and \*\*leave-one-out\*\* algorithm as cross-validation. I would like to make research step-by-step from basic to complicated model

\*

## Conclusion

There are two problem in this data.

\* The return of I2 index at the 2015-03-25 is oulier.

\* \*\*"Zero-return"\*\* problem. The presence of $0$ in the data tend to the ambiguity of the betas choice. In this data where are a lot of zeros for the cash-like asset.

x

**# The case study: "No maximum at the R2-curve"**

**## Author**

Olga Krasotkina for MPI

​

**## Problem statement**

In some cases the PR2 curve is monotonic and has not any extremum and sugessts very large optimal smoothness tending to the full stationary betas.

​

**## Source data**

​

There are 12 indicies an 158 funds for 60 time points from 2015-03-29 till 2015-06-12.

​

​

**## Research steps**

​

\* Done Python workspace for non-stationaty regression and cross-validation analysis. At this stage I use the non-stationaty regression estimation procedure without any constraints and **\*\*leave-one-out\*\*** algorithm as cross-validation. I would like to make research step-by-step from basic to complicated model

\*

​

**## Conclusion**

​

There are two problem in this data.

​

\* The return of I2 index at the 2015-03-25 is oulier.

\* **\*\*"Zero-return"\*\*** problem. The presence of $0$ in the data tend to the ambiguity of the betas choice. In this data where are a lot of zeros for the cash-like asset.

**Extended studies¶**

**Step 1. Import packages¶**

In [22]:

**import** pandas **as** pd

**import** numpy **as** np

**import** scipy **as** sc

**import** Regress **as** nsr

**import** matplotlib.pylab **as** plt

**from** datetime **import** datetime

In [23]:

**%**matplotlib notebook

**Step 2. Import data¶**

In [24]:

fund\_data = pd.read\_excel('SmoothnessData.xlsx','Sheet1')

assets\_data = pd.read\_excel('SmoothnessData.xlsx','Sheet2')

*#fund\_data*

In [25]:

funds = fund\_data.iloc[2:62,0:158].values

assets = assets\_data.iloc[2:62,0:12].values

dates = assets\_data.iloc[2:62].index.date

​

In [26]:

n = np.size(assets,1)

T = np.size(assets,0)

**print**(n)

**print**(T)

12

60

**Step 3. A look at the returns¶**

In [27]:

**from** matplotlib.ticker **import** MultipleLocator

majorLocator = MultipleLocator(5)

​

assets = np.mat(assets)

funds = np.mat(funds)

time = range(T)

plt.figure(1, figsize=(10,5))

plt.title('Indicies returns')

plt.xticks(time, dates, rotation=90, size = 7)

ax = plt.gca()

ax.get\_xaxis().set\_tick\_params(pad=5)

​

​

ax.get\_xaxis().set\_major\_locator(majorLocator)

​

*#plt.subplot(3,1,1)*

**for** i **in** range(0,n):

plt.plot(time, assets[:,i])

plt.show()

​

**Figure 1**

Download plot

As you can see the return data series are correlated. But the I2 index at the 2015-03-25 has oulier.

**Step 3. Example of an non-stationary regression¶**

In [28]:

time = range(T)

plt.figure(2,figsize=(10,5))

x

*# Smoothness*

lam = 50

​

assets = np.mat(assets)

funds = np.mat(funds)

fund = funds[:,0]

outPoint = np.zeros([T,1])

beta = nsr.nonstatRegress(assets, fund, lam, outPoint)

​

​

time = range(T)

plt.figure(2,figsize=(10,5))

plt.title('Betas')

plt.xticks(time, dates, rotation=90, size = 7)

ax = plt.gca()

ax.get\_xaxis().set\_tick\_params(pad=5)

ax.get\_xaxis().set\_major\_locator(majorLocator)

​

​

*#plt.subplot(3,1,2)*

**for** i **in** range(1,n):

plt.plot(time, beta[:,i])

​

plt.show()

**Figure 2**

x= y=0.397568

**Step 4. Example of cross-validation procedure for fixed smoothness value¶**

In [29]:

*# set smoothness*

lam = 100

*# set a fund to be analyzed*

fund = funds[:,2]

*# Cross-validation (Leave-one\_out procedure)*

arrR2, r2 = nsr.LeaveOneOut(assets,fund,lam)

*# Plot PR2*

time = range(T)

plt.figure(3,figsize=(10,5))

plt.title('Predicted R2 at different time point and fund f2 ')

plt.xticks(time, dates, rotation=90, size = 7)

ax = plt.gca()

ax.get\_xaxis().set\_tick\_params(pad=5)

ax.get\_xaxis().set\_major\_locator(majorLocator)

plt.plot(time, arrR2)

plt.show()

**Figure 3**

In [30]:

for all funds

*### Step 5. Predicted R2 for all funds*

In [31]:

*# set smoothness*

lam = 100

​

*# Cross-validation (Leave-one\_out procedure)*

arrR2, r2 = nsr.LeaveOneOut(assets,fund,lam)

*# Plot PR2*

time = range(T)

plt.figure(4,figsize=(10,5))

plt.title('Predicted R2 at different time point and fund f2 ')

plt.xticks(time, dates, rotation=90, size = 7)

ax = plt.gca()

ax.get\_xaxis().set\_tick\_params(pad=5)

ax.get\_xaxis().set\_major\_locator(majorLocator)

**for** i **in** range(10):

arrR2, r2 = nsr.LeaveOneOut(assets,funds[:,i],lam)

plt.plot(time, arrR2)

plt.show()

**Figure 4**

x= y=59.7422

In [19]:

**print** r2

0.544710163579

In [20]:

assets = np.mat(assets)

funds = np.mat(funds)

fund = funds[:,1]

lam\_arr = np.arange(100,300,20)

r2\_lam\_arr = []

arr = []

**for** lam **in** lam\_arr:

arr, r2\_lam = nsr.LeaveOneOut(assets,fund,lam)

r2\_lam\_arr.append(r2\_lam)

*#print r2\_lam\_arr*

plt.figure(4)

plt.plot(lam\_arr,r2\_lam\_arr)

*#plt.show()*

**Unrecognized output: undefined**

Out[ ]:

Out[20]:

[<matplotlib.lines.Line2D at 0x1093b42d0>]

In [ ]:

​